## REMARKS/ARGUMENTS

Prior to this Amendment, claims 1-18 were pending. In this Amendment, claims 1 and 2 are amended to clarify that the client object is generated at the client machine and is configured to implement the remotely accessible methods on the client machine rather than simply performing remote method invocation where the method is implemented on the remote or server device.

Independent claim 7 is amended to clarify that the target object on the remote station is modified or managed based on actions at the client station, and also, to include the limitations of now canceled claim 10 to clarify that the client object is not merely a set of remote interfaces for a remote object. Claim 11 is amended to correct dependency.

Claims 15-18 are canceled.

No new matter is added by the claim amendments, and claims 1-9 and 11-14 remain for consideration by the Examiner.

### Claim Rejections Under 35 U.S.C. §102

In the February 23, 2004 Office Action, claims 1, 2, 5, 7, 8, and 15-18 were rejected under 102(b) as being anticipated by "A Distributed Object Model for the Java™ System" ("Wollrath"). Claims 15-18 are canceled by this Amendment. The rejection of claims 1, 2, 5, 7, and 8 is traversed based on the amendments to claims 1, 2, and 7 and the following remarks.

Initially, it may be useful to summarize several features of the invention as discussed in Applicants' specification at line 8, page 34. The present invention is directed toward a robust method and system for remotely managing distributed objects – not simply for remotely invoking distributed methods. In this regard, a typical method involves registering one or more managed objects or beans with a framework, such as with a registry service of the framework, registering one or more network adapters providing network communications protocols with the framework, and then enabling external access via a network to the framework and

the registered managed objects or beans. At a client station remote to th framework implementation, a client object is generated forming a representation of one of the registered and now, targeted objects or beans. This generating typically includes compiling the target object on the client station and generating the client object which comprises a target object interface identifying which methods of the target object are remotely accessible and manageable and a target object stub that implements the target methods on the client station. Hence, the client management application is able to access and manage the target object or bean by instantiating it on the client station. Management may include, as discussed at page 15, lines 10-24, the extraction of bean properties such as through introspection of the implemented client object, modifying aspects of the bean or object, such as by changing properties, that are then communicated to the remote station, such as through GET, SET, and POST responses.

Claim 1 is directed to a method for managing from a client machine a target object at a remote station. The method includes generating a client object forming a representation of the target object on the client machine. The generated client object is configured so as to extract the methods that are remotely accessible and manageable (i.e., "support manipulation of properties of said target object") and to implement such remotely accessible methods on the client machine. Further, the method calls for "registering" the target object with a framework at the remote station and then, enabling a client application to access the methods at the client machine "which support remote manipulation by instantiating the client object." The rejection of claim 1 is not supported by the cited reference because Wollrath fails to teach or even suggest each of these features.

Applicant disagrees with the interpretation of Wollrath presented in the Office Action. Wollrath is directed to a distributed object model for remotely invoking distributed Java objects (see Abstract, "We have designed such a model and implemented a system that supports remote method invocation (RMI) for distributed objects in Java."). Wollrath fails to teach the method of claim 1 which calls for a

target object to be implemented as a client object and for the methods to be implemented on the client machine – not just invoked – and to also manipulate the instantiated client objects to manage the target object. As discussed in col. 2 of page 225, a "client invoking a method on a remote server object actually makes use of a stub or proxy for the remote object **as a conduit to the remote object**."

The Wollrath "stub" is not the generated client object of claim 1. Instead, the "stub is an implementation of the remote interfaces of the remote object and forwards invocation requests to that server object via the remote reference layer." As defined at col. 2, page 220, an "interface, in Java, describes a set of methods for an object, but provides no implementation." Hence, Wollrath describes using a stub for providing remote interfaces to distributed or remote methods run or implemented on other stations or machines but fails to describe extracting, such as through introspection, methods in a registered target object that can be remotely manipulated or managed and then generating a client object on the client machine which implements those particular methods and facilitates modification or management of the target object still on the remote station.

The Office Action cites Wollrath at page 228, second column for teaching some of the features of claim 1. As discussed earlier, at this citation, the stub code is taught as providing a set of remote interfaces, but the stub code is not an implementation of the methods of a target object (which are instead later invoked via the interfaces of the stub code). Additionally, Wollrath fails to teach that the stub extracts methods that of the target object that "support manipulation of properties" as called for in claim 1. This feature allows the target object to be managed not merely called with input parameters as suggested by the Office Action's reference to "remote procedure calls" and invocations.

Wollrath does not teach that the target objects are registered with a framework. The Office Action cites page 227, first and second columns of Wollrath, but at this citation, Wollrath is describing setting up connections and common transports. Wollrath at this citation and elsewhere does not teach the usefulness of

registering one or more target objects, e.g., managed objects or beans, with a framework, such as with a registry service to facilitate location of such manageable or managed objects (and in some cases, registration is a prerequisite for management as described in Applicants' specification at page 22, line 20). Further, Wollrath does not teach enabling a client application (such as a web browser) to access the managed methods at the client machine "which support remote manipulation" by instantiation of the client object on the client machine. Again, this feature allows the method to be used to manage methods of a target object via a client object generated at the client machine. In light of the above remarks, the rejection of claim 1 is not supported by Wollrath and withdrawal of the rejection is respectfully requested.

Claims 2 and 5 depend from claim 1 and are allowable at least for the reasons for allowing claim 1. Additionally, claim 2 calls for the generation of the client object at the client machine to involve compiling the target object including implementing the remotely accessible methods. The Office Action cites Wollrath at page 228, column 2 for teaching this element. At this citation, Wollrath describes downloading stub code onto the client machine, which would provide remote interfaces but would not provide implementations of methods that are accessible for remote manipulation. As discussed with reference to claim 1, Applicant can find no support for the assertion that the stubs of Wollrath are equivalent to the claimed client objects (or target objects). Further, even if the Wollrath stubs were created by compilation, Wollrath would fail to teach compiling target objects to generate the client objects on the client machine as called for in claim 2. For this additional reason, the rejection based on Wollrath is improper and should be withdrawn.

Independent claim 7 is directed to a mechanism at a client station for accessing and modifying a target object at a remote station. Claim 7 includes several limitations similar to that of claim 7 and as a result, many of the arguments provided for claim 1 are equally applicable to claim 7. Additionally, claim 7 specifies that the target object be modified on the remote station by a client application on a

client station via access to a local client object. The client object comprises a set of properties, a set of methods for performing actions, and support for events and for introspection.

Wollrath teaches invoking methods in a remote object but does not teach modifying the remote object (such as by altering the properties, methods, or support) via a client object on a client station. The Office Action refers to the Wollrath client stub, but as discussed with reference to claim 1, this stub is not the client object described in Applicants' specification and/or called for in claim 7. Wollrath provides a useful system for remotely invoking objects on distributed devices but fails to teach how such remote objects can be managed and their properties and methods be modified via a client object generated on a client station. As a result, Wollrath does not support the rejection of claim 7, and withdrawal of the rejection is respectfully requested.

Claim 8 depends from claim 7 and is believed allowable as depending from an allowable base claim. Additionally, claim 8 includes limitations similar to claim 2 and the reasons provided for allowing claim 2 over Wollrath are equally applicable to the mechanism of claim 8.

**Claim Rejections Under 35 U.S.C. §103**

Additionally, in the February 23, 2004 Office Action, claims 3 and 9 were rejected under 103(a) as being unpatentable over Wollrath in view of "Specializing Object-Oriented RPC for Functionality and Performance" ("Zelesko"). This rejection is traversed based on the following remarks.

Claims 3 and 9 depend from allowable claims 1 and 7, respectively, and each is believed allowable as depending from an allowable base claim. Zelesko does not overcome the shortcomings of Wollrath with relation to claims 1 and 7, and a rejection is not supported by the combination of these two references. As to claims 3 and 9, the additional limitation of selectively replacing a target object stub at the client machine (e.g., in the client object that is generated on the client

machine) is performed "for dynamically modifying the behaviour of said client application at runtime." The Office Action cites Zelesko at page 176 first and second columns for teaching this feature. However, Zelesko is directed toward an improved method of implement remote procedure call (RPC) techniques and fails to teach, as called for in claim 1, the generating of a client object on a client machine and then modifying a portion of that instantiated object, i.e., the target object stub, so as to modify behavior of a client application accessing the client object. The target object stub, while including the term "stub", is not equivalent to the "stub" discussed in Zelesko. As with Wollrath, the Zelesko stub is a proxy used to provide an interface to the remote object so as to facilitate procedure calls but it does not implement the remote object on a client machine. As a result, the combination of Wollrath and Zelesko fails to teach or suggest the claimed method and mechanism, and the rejections of claims 3 and 9 are not supported and should be withdrawn.

Further, in the February 23, 2004 Office Action, claim 4 was rejected under 103(a) as being unpatentable over Wollrath in view of "IBM Visualage for Java**" ("the IBM reference"). This rejection is traversed because claim 4 depends from claim 1 and is allowable as depending from an allowable base claim. The IBM reference does not overcome the deficiencies of Wollrath discussed with reference to claim 1. Further, there is no motivation in Wollrath to modify its teachings to generate a client object on the client machine based on a target object, as Wollrath is directed to using RMI for distributed objects. The IBM reference does discuss Java beans, but when combined with the teachings of Wollrath, the combination does not produce the invention claimed in claim 1 or dependent claim 4 and does not support a proper rejection under 103(a).

The Office Action further rejected claim 6 under 103(a) as being unpatentable over Wollrath in view of EP 0727739 A1 ("Hollberg"). Claim 6 depends from claim 1 and is allowable because it depends from an allowable base claim. Additionally, Hollberg fails to overcome the deficiencies of Wollrath

discussed with reference to claim 1. Hence, the combination of Wollrath and Hollberg does not teach or support a rejection of the method of claim 1 or claim 6.

The Office Action further rejected claims 10-12 under 103(a) as being unpatentable over Wollrath in view of Zelesko further in view of the IBM reference. Claim 10 is canceled by the amendment with its limitations being added to independent claim 7. Claims 11 and 12 depend from claim 7. Zelesko combined with the IBM reference teaching does not overcome the deficiencies in Wollrath discussed earlier with reference to claim 7. Hence, claims 11 and 12 are believed allowable for at least the reasons provided for allowing claim 7.

**Conclusion**

In view of all of the above, the claims are now believed to be allowable and the case in condition for allowance which action is respectfully requested. Should the Examiner be of the opinion that a telephone conference would expedite the prosecution of this case, the Examiner is requested to contact Applicants' attorney at the telephone number listed below.

No fee is believed due for this submittal. However, any fee deficiency associated with this submittal may be charged to Deposit Account No. 50-1123.

Respectfully submitted,

May 21, 2004

Kent A. Lembke, No. 44,866
Hogan & Hartson LLP
One Tabor Center
1200 17th Street, Suite 1500
Denver, Colorado 80202
(720) 406-5378 Tel
(303) 899-7333 Fax